

```

from os.path import exists

# Initialize WORDS as an empty array
WORDS = []

# A function to reduce redundant code
def invalid_error():
    print("Invalid Input")

# A function to take confirmation from user
def is_confirmed(text):
    decision = input(f"{text} (y,n) : ")
    if decision.lower() == "y":
        return True
    return False

# Initialize words from a text file to the program
def init():
    if exists("ReservesDictionary/data.txt"):
        data_file = open("ReservesDictionary/data.txt", "r")
        data = data_file.readlines()
        for line in data:
            if len(line.split("|")) == 3:
                name, description, sample = line.strip().split("|")
                WORDS.append(
                    {
                        "name": name,
                        "description": description,
                        "sample": sample,
                    }
                )
        data_file.close()

# Function to save all words in a text file
def save_file():
    data_file = open("ReservesDictionary/data.txt", "w")
    data_file.writelines(
        f'{"|".join((str(item[key]) for key in item))}\n' for item in WORDS
    )
    data_file.close()

# Take input for a word on adding and updating
def take_word_inputs():
    name = input("Name : ")
    description = input("Description : ")
    sample = input("Sample : ")
    return {"name": name, "description": description, "sample": sample}

# Function to add new word
def add_word():
    print("New word :")
    WORDS.append(take_word_inputs())
    print_all_words()
    if is_confirmed("Add again ?"):
        add_word()

```

```

# Function to update an existing word
def update_word():
    print_all_words()
    option = int(input("Enter a number to update : "))
    if option <= len(WORDS):
        WORDS[option - 1] = take_word_inputs()
        print_all_words()
        if is_confirmed("Update again?"):
            remove_word()
    else:
        print("Invalid input")

# Print words from arguments to the console
def print_words(words):
    if len(words) == 0:
        print("\n\nNo word found\n\n")
        return
    index = 0
    for word in words:
        index += 1
        print("\n", index, " .")
        print("Name : ", word["name"])
        print("Description : ", word["description"])
        print("Sample : ", word["sample"], "\n")

# Print all words to the console
def print_all_words():
    print_words(WORDS)

# Search for a word
def search_word(query):
    results = []
    for item in WORDS:
        if (
            item["name"].lower().__contains__(query.lower())
            or item["description"].lower().__contains__(query.lower())
            or item["sample"].lower().__contains__(query.lower())
        ):
            results.append(item)
    return results

# Function to print search result to the console
def show_search_result():
    query = input("Search for word : ")
    print_words(search_word(query))

# Function to remove existing word
def remove_word():
    print_all_words()
    option = int(input("Enter a number to remove : "))
    if option <= len(WORDS):
        del WORDS[option - 1]
        print_all_words()
        if is_confirmed("Delete again?"):
            remove_word()
    else:
        print("Invalid input")

```

```

# Main function containing the menu of the application
def main():
    init()
    while True:
        print(
            """

            Welcome to Reserve Dictionary

            1. View words
            2. Add New word
            3. Update word
            4. Remove word
            5. Search word
            0. Exit
            """
        )
        option = int(input("Enter an option : "))
        if option == 0:
            break
        menus = {
            1: print_all_words,
            2: add_word,
            3: update_word,
            4: remove_word,
            5: show_search_result,
        }
        if menus.keys().__contains__(option):
            menus[option]()
        else:
            print("Invalid Option")
    save_file()

main()

```